Hans Zantema
Faculteit Wiskunde & Informatica
Technische Universiteit Eindhoven
Postbus 513
5600 MB Eindhoven
H.Zantema@tue.nl

Event International Mathematical Olympiad 2010

A problem from IMO 2010

Every year there is an International Mathematical Olympiad (IMO) for high school students; in 2010 it was held in Kazakhstan, with 517 participants from 97 countries. As every year, they got six problems to solve. Problem 5 of IMO 2010 was proposed by Hans Zantema. He was inspired by some observations in his research in theoretical computer science. This paper presents some of this background, and includes a full solution.

Problem 5 of IMO 2010 is formulated as follows:

Problem 5. In each of six boxes $B_1, B_2, B_3, B_4, B_5, B_6$ there is initially one coin. There are two types of operation allowed:

Type 1: Choose a nonempty box B_j with $1 \le j \le 5$. Remove one coin from B_j and add two coins to B_{j+1}.

Type 2: Choose a nonempty box B_k with $1 \le k \le 4$. Remove one coin from B_k and exchange the contents of (possibly empty) boxes B_{k+1} and B_{k+2}.

Determine whether there is a finite sequence of such operations that results in boxes B_1, B_2, B_3, B_4, B_5 being empty and box B_6 containing exactly $2010^{2010^{2010}}$ coins. (Note that $a^{b^c} = a^{(b^c)}$.)

The Ackermann function

Some functions grow faster than others. Restricting to functions from natural numbers to natural numbers, by extremely simple means one can construct a wide range of functions, some of which turn out to grow extremely rapidly. A basic building block is the *successor function* s mapping every x to its successor $s(x) = x + 1$. A fresh function f may be defined *recursively* by defining $f(0)$, and expressing $f(s(x))$ as a function in $f(x)$. For instance, one can define the function double mapping x to $\mathrm{double}(x) = 2x$ by

$$\mathrm{double}(0) = 0$$
$$\mathrm{double}(s(x)) = s(s(\mathrm{double}(x))).$$

Next, one can define the function exp mapping x to $\exp(x) = 2^x$ by

$$\exp(0) = s(0)$$
$$\exp(s(x)) = \mathrm{double}(\exp(x)).$$

A very fast growing function power can be obtained by defining

$$\mathrm{power}(0) = s(0)$$
$$\mathrm{power}(s(x)) = \exp(\mathrm{power}(x)),$$

representing $\mathrm{power}(x) = 2^{2^{2^{\cdots}}}$, a tower of height x. So for instance, $\mathrm{power}(5) = 2^{65536}$, being a number having over 19,000 digits in decimal notation, probably being much more than the number of atoms in the universe. Functions that are composed in this way are called *primitive recursive*. More precisely, the class of primitive recursive functions is defined to be the smallest class of functions from tuples of natural numbers to natural numbers such that

- the constant zero function and the successor function s are primitive recursive;
- the class is closed under projection and composition;
- if f and g are primitive recursive, then h defined by

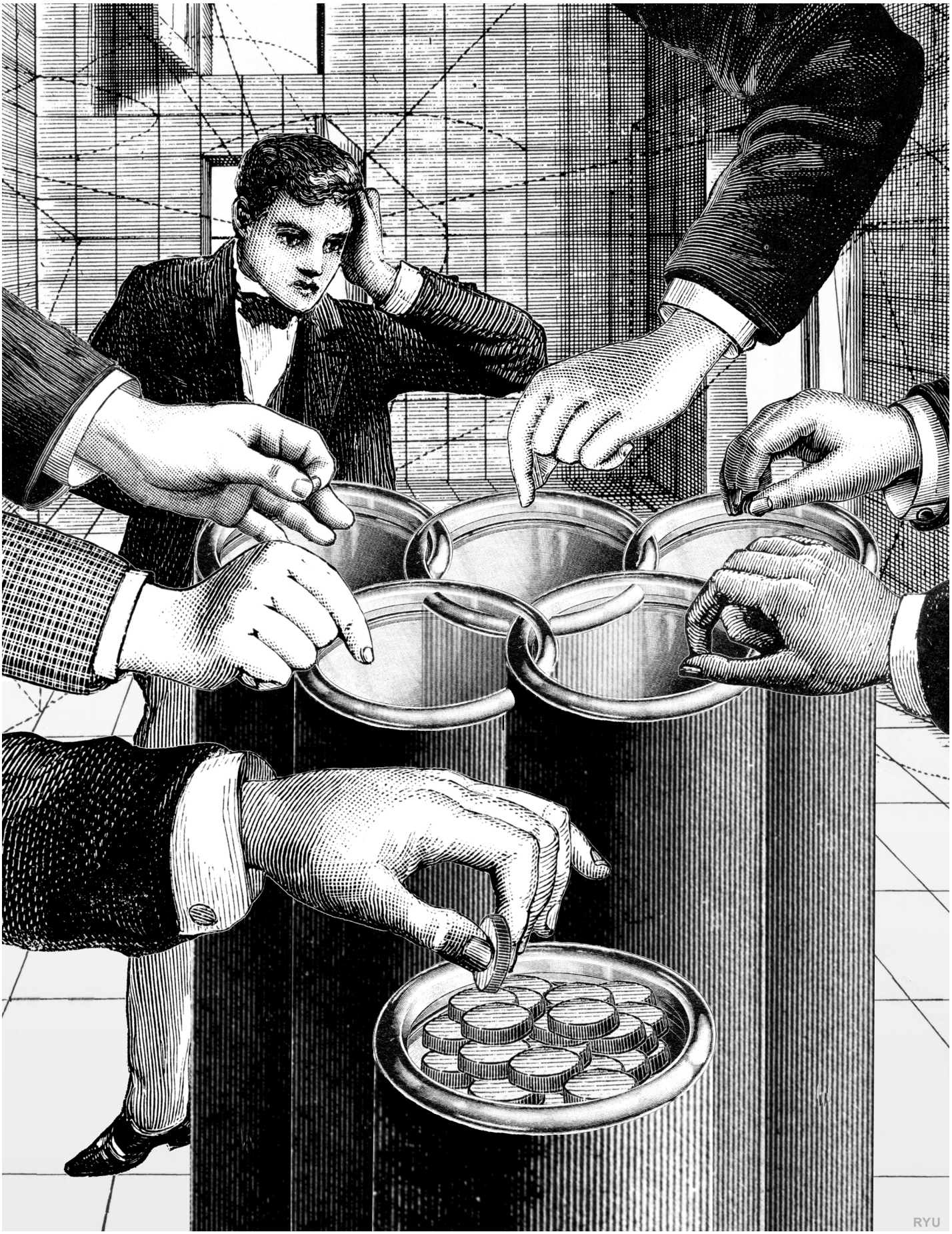
$$h(0, x_1, \ldots, x_k) = f(x_1, \ldots, x_k)$$
$$h(s(y), x_1, \ldots, x_k) = g(y, h(y, x_1, \ldots, x_k), x_1, \ldots, x_k)$$

  is primitive recursive too.

As our function definitions for double, exp and power all follow this pattern of primitive recursion, we see that some primitive recursive functions like power already grow extremely fast. A fundamental question now is the following. Is it possible to define functions recursively, but not following the format of primitive recursion, that grow even faster? The answer is: yes, this is possible. A standard example is the *Ackermann function* A defined by

$$A(x,y) = \begin{cases} y+1 & \text{if } x = 0 \\ A(x-1,1) & \text{if } x > 0 \text{ and } y = 0 \\ A(x-1, A(x, y-1)) & \text{if } x > 0 \text{ and } y > 0. \end{cases}$$

Although on a first view this function definition looks simple and innocent, very small arguments already give amazingly high values. For instance, $A(4,2) = \mathrm{power}(5) - 3$ again has over 19,000 digits in decimal notation, while $A(5,1) = \mathrm{power}(65533) - 3$ is much larger. It was conjectured by David Hilbert in the 1920's that a three argument variant of this function is not primitive recursive. This was proved in 1928 by his student Wilhelm Ackermann, after whom the function has been



Illustratie: Ryu Tajiri

named. The current binary version A was proposed later by Péter and Robinson, but is usually called the Ackermann function nowadays.

Relation to the IMO problem

But how does this ancient theory relate to the above mentioned IMO problem? It turns out that after generalizing the number of boxes to arbitrary numbers, the game described in the problem can mimic functions closely related to the Ackermann function. More precisely, we will show now that starting from a sequence of $2x+1$ boxes of which the leftmost contains $y+1$ coins and the others are empty, by applying the operations of the two types we can reach the configuration in which the rightmost box contains $f(x,y)$ coins and the others are empty. Here f is the function exceeding the Ackermann function defined by

$$f(x,y) = \begin{cases} y+1 & \text{if } x = 0 \\ f(x-1,3) & \text{if } x > 0 \text{ and } y = 0 \\ f(x-1,f(x,y-1)+4) & \text{if } x > 0 \text{ and } y > 0. \end{cases}$$

In order to do so we start by giving some notation. Write $[a_1,a_2,a_3,\ldots]$ for a sequence of boxes, containing $a_1,a_2,a_3,\ldots$, respectively, from left to right. So the type 1 operation states that in such a sequence two consecutive numbers a,b may be replaced by $a-1,b+2$ in case $a>0$, and the type 2 operation states that in such a sequence three consecutive numbers a,b,c may be replaced by $a-1,c,b$ in case $a>0$. We prove the claim by induction on x. For $x=0$ it is trivial by doing no steps at all and using $f(0,y)=y+1$. For $x>0$ we apply induction on y. For $y=0$ we first replace the initial configuration $[1,0,0,\ldots]$ by $[0,0,4,0,\ldots]$ by doing three steps of type 1, and then apply the outer induction hypothesis yielding the desired number $f(x-1,3)=f(x,0)$ coins in the rightmost box. For $y>0$ we keep one coin in the leftmost box and apply the inner induction hypothesis to replace the initial configuration $[1,0,0,\ldots]$ by

$$[1,0,0,\ldots,0,f(x,y-1)]$$

followed by applying steps of type 1 yielding

$$[0,1,1,\ldots,1,0,f(x,y-1)+4].$$

Next we apply steps of type 2 for each of the 1's, executed from right to left, yielding

$$[0,0,f(x,y-1)+4,0,0,\ldots,0],$$

from which the outer induction hypothesis yields a replacement to the desired configuration in which the last box contains $f(x-1,f(x,y-1)+4)$ coins, concluding the proof.

Note that in this construction steps of type 2 are only applied if the leftmost of the two exchanging boxes is empty.

Knowing that by the relation to the Ackermann function extremely high numbers of coins can be achieved from simple initial configurations, it is a natural question what is the smallest number of boxes for which this extreme blow-up shows up. It turned out that this already occurs for six boxes. Moreover, it turned out that these investigations could be given completely elementary, by which the idea was born to propose this as an IMO problem. It is a kind of tradition that if high numbers occur in an IMO problem, an instance of a high number is chosen reflecting the year. So high numbers occurring in IMO 2010 should refer to the number 2010. An obvious choice now was 2010^{2010}. In my original proposal for the IMO problem the question was to make this number 2010^{2010}; later on this was extended to $2010^{2010^{2010}}$. Another difference with my original proposal is that in the eventual problem it is asked whether the particular configuration can be reached, while in my original proposal it was asked to prove that it can be reached, so already including the information that it can be reached indeed. An issue from my original version that has been kept is the fact that this value has to be reached exactly.

Now we present an elementary solution for the problem not referring to the above relationship with the Ackermann function.

A Solution

Indeed, the intended final configuration can be reached. Write $M = 2010^{2010^{2010}}$. We write $\to_1$ for doing a type 1 step, $\to_2$ for doing a type 2 step, and $\to^*$ for doing any number of steps. So we have to prove that $[1,1,1,1,1,1] \to^* [0,0,0,0,0,M]$.

We will use the function power as introduced above by $\mathrm{power}(0)=1$ and $\mathrm{power}(x+1)=2^{\mathrm{power}(x)}$ for $x \geq 0$, so $\mathrm{power}(x) = 2^{2^{2^{\cdots}}}$, containing x copies of 2. First we derive a bound on M in terms of power.

Since $2010^{2010} < (2^{11})^{2010} = 2^{22110} < 2^{2^{15}}$, we obtain

$$M = 2010^{2010^{2010}} < (2^{11})^{2^{2^{15}}} = 2^{11*2^{2^{15}}} < 2^{2^{2^{16}}} = \mathrm{power}(6).$$

In the sequel, k,n are arbitrary numbers ≥ 0. Starting from $[n+1,k,0]$ we can do k steps of type 1 yielding $[n+1,0,2k]$, so

$$[n+1,k,0] \to^* [n+1,0,2k] \to_2 [n,2k,0]. \tag{1}$$

Starting from $[n+1,0,0] \to_1 [n,2,0]$, applying (1) exactly n times yields $[0,2^{n+1},0]$, so

$$[k+1,n+1,0,0] \to^* [k+1,0,2^{n+1},0] \to_2 [k,2^{n+1},0,0]. \tag{2}$$

Now starting by $[k+1,0,0,0] \to_1 [k,2,0,0]$ and then apply (2) exactly k times yields

$$[k+1,0,0,0] \to^* [0,\mathrm{power}(k+1),0,0]. \tag{3}$$

In order to apply the key observation (3) we first have to make the last three numbers equal to zero, and in front of it a number that is at least 6. One way to do so is

$$\begin{aligned} [1,1,1,1,1,1] &\to_1 [1,1,1,1,0,3] \to_2 [1,1,1,0,3,0] \\ &\to_2 [1,1,0,3,0,0] \to_2 [1,0,3,0,0,0] \\ &\to_1 [0,2,3,0,0,0] \to_1^* [0,0,7,0,0,0]. \end{aligned}$$

Now (3) yields $[0,0,0,\mathrm{power}(7),0,0]$. Next, applying $\mathrm{power}(7)-M/4$ steps of type 2 yields $[0,0,0,M/4,0,0]$, followed by type 1 steps yielding the desired end configuration $[0,0,0,0,0,M]$, concluding the proof.

This construction allows several variations, some of which yielding values much greater than M, for instance,

$$\begin{aligned} &[1,1,1,1,1,1] \to_1 [1,1,0,3,1,1] \to_1 [1,1,0,2,3,1] \\ &\to_1^* [1,1,0,2,0,7] \to_2 [1,1,0,1,7,0] \to_1^* [1,1,0,1,0,14] \\ &\to_2 [1,1,0,0,14,0] \to_1 [0,3,0,0,14,0] \to_1 [0,2,2,0,14,0] \\ &\to_2 [0,2,1,14,0,0] \to^* [0,2,1,0,2^{14},0] \to_2^* [0,1,2^{14},0,0,0] \\ &\to^* [0,1,0,\mathrm{power}(2^{14}),0,0] \to_2 [0,0,\mathrm{power}(2^{14}),0,0,0] \\ &\to^* [0,0,0,\mathrm{power}(\mathrm{power}(2^{14})),0,0]. \end{aligned}$$

Then similar as above for any number N divisible by 4 satisfying $N/4 \leq$ power(power(2^{14})), the configuration $[0,0,0,0,0,N]$ can be obtained, for instance for $N = 2010^{2010^{2010^{\cdots}}}$, being a tower of height 2010, or of height M.

Relation to rewriting

We saw that it is possible to start in a configuration with only a few boxes and a few coins, and end up in a configuration with an amazing high number of coins in the rightmost box. On the other hand, the game can not go on forever, since in every step the sequence of numbers lexicographically decreases, and the lengths of the sequences are fixed. Such a game that can not go on forever is called *terminating*. In the past years I did a lot of research in proving termination of computation, in particular in a standard format describing computation called *rewriting*. In this area a game like this is of particular interest: it is easy to describe, it is terminating, but it allows computations of extremely high numbers of steps. Several encodings of this game in rewriting were described already years ago. For instance, in [1] Hofbauer and Lautemann gave an encoding as a term rewriting system, and gave a similar argument as we did for showing the relationship with the Ackermann function. In [2] Touzet gave an encoding as a string rewriting system. A simpler string rewriting system with the same property is the following:

$$ab \to baa,\ abb \to bc,\ ca \to ac,\ c \to b.$$

This means that the strings to be rewritten are finite sequences only composed from the three symbols a, b, c. The rules state that any occurring pattern ab may be replaced by baa, any pattern abb by bc, and so on. For instance, one can rewrite

$$abba \to bca \to bac \to bab \to bbaa.$$

Modern termination provers like AProVE or TTT2 easily prove termination of this string rewriting system fully automatically. On the other hand, rewrite sequences may be very long. Writing a^n for n consecutive copies of a, and encoding $[n_1, n_2, n_3, n_4, n_5, n_6]$ by $a^{n_1}ba^{n_2}ba^{n_3}ba^{n_4}ba^{n_5}ba^{n_6}$, we see that a type 1 step can be mimicked by an application of the first rule $ab \to baa$. Further, a type 2 step replacing $n+1, 0, k$ by $n, k, 0$ can be mimicked by first an application of the rule $abb \to bc$, then k applications of the rule $ca \to ac$, and finally an application of the rule $c \to b$. For instance, for $k = 2$ we have

$$abbaa \to bcaa \to baca \to baac \to baab.$$

In the given solution of the IMO problem all type 2 steps were of the shape replacing $n+1, 0, k$ by $n, k, 0$. As a consequence of this solution, using these four rewrite rules it is possible to start in $ababababaaba$ and end in $bbbbba^M$, being a string ending in $M = 2010^{2010^{2010}}$ copies of the symbol a. As in every rewrite step the length of the string increases by at most one, the number of rewrite steps to reach this final string is more than M. Using our argument relating the game to the Ackermann function shows that it is possible to make computations of which the size of the final string, and hence also the number of rewrite steps, is a non-primitive recursive function in the size of the initial string. Currently this four rule string rewriting system is the smallest known terminating string rewriting system of which the computation length is non-primitive recursive in the size of the initial string.

Concluding remarks

Even after understanding the solution of the problem, it remains amazing that by these extremely simple rules such unwieldy high values can be obtained, starting in such a small initial configuration. The particular game and its Ackermann function like behavior were already well-known. My contribution mainly restricted to some investigations for small initial configurations, the formulation of the problem and passing it to the source of potential IMO problems. As every year only six problems are chosen from a resource of over 100 proposals, it is an honor if a particular problem is chosen. In this way the hundreds of IMO participants, being promising future scientists from around 100 countries, have become in touch with these remarkable observations. As a participant of IMO 1974, after having been the winner of its preceding Dutch Mathematical Olympiad, for me personally it was a great experience to play such a complementary role in the IMO of 36 years later.

In my current research in computer science I see how the mathematical way of working by abstraction and giving formal proofs does not restrict to traditional mathematics, but is also the basis of theoretical computer science. Along this line I am happy that the IMO does not restrict to problems from traditional mathematics, but also presented this problem with such a computational flavor.

www.imo2010org.kz

References

1 D. Hofbauer and C. Lautemann. 'Termination proofs and the length of derivations', in N. Dershowitz, editor, *Rewriting Techniques and Applications, 3rd International Conference, RTA'89*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177. Springer, 1989.

2 H. Touzet. 'A complex example of a simplifying rewrite system', in K. Larsen, S. Skyum, and G. Winskel, editors, *Automata, Languages and Programming, 25th International Colloquium, ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 507–517. Springer, 1998.